Linux in Large Installations

Using Linux in Large, Widespread, Mission-Critical Environments

Damian Ivereigh

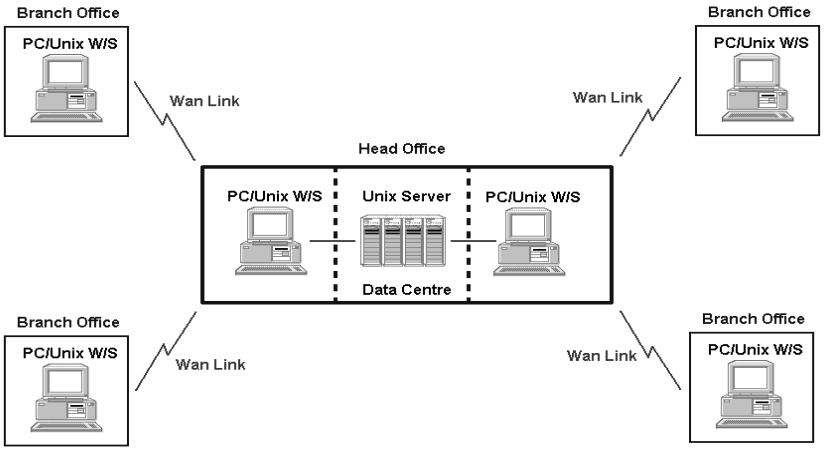
damian@tpp.org

http://www.tpp.org/CiscoPrint

Overview

- Discuss unique problems of large installations
- Traditional methods of tackling the problem
- Show how Linux is a good position to tackle these problems is a new way
- Tips and lessons learnt doing it

Traditional Approach - Centralisation



Advantage of centralisation

- Administration centralised
- Administration & security simple
- Updates easy (only one machine)
- Spend lots of money on failure prevention - UPS, mirror machines etc

Problems with centralisation

- Very dependant on WAN performance.
- Central machine can be single point of failure.
- Cannot expand past a certain size - you then have to start having multiple systems.
- Multiple central machines lose all the advantages of simplicity.

Get the best of both worlds?

- Can we have multiple machines, yet still consider it to be a single machine?
- Can we put these machines on other side of WAN links?
- Can we make it simple to administer?
- Can we make it reliable?

Multiple Machines - Linux!

- May as well go the whole hog - have lots of small machines spread all over the place.
- Must be cheap - Linux
- Must be remotely administered - Linux
- Must be upgradeable remotely - Linux
- Must be reliable - Linux

Network Directory Service

- Store all data, including configuration data in a directory service.
- Every machine **exactly** the same - use bootp or dhcp as well.
- On boot up, each machine consults directory to decide how to behave, what to service etc.

Global Machine

- Now have each machine look at a global configuration as held in the NDS.
- Each machine will then play its part in this configuration.
- A single change in the NDS can make all the machines behave differently.

Example - Print System at Cisco

- Each printer created centrally in the NDS. Holds all IP, model and communication info.
- Also stored on each printer record is the allocated print server.
- Any other print server that receives a print job, simply forwards that job to the correct print server.

How? /etc/printcap

- Standard /etc/printcap can direct LPD to either:
  - Print to a printer through a script
  - Send the job to another server.
- Change the backend routine in LPD that usually reads the /etc/printcap file to now construct the entry from the NDS.

Fault tolerant - failover

- Instead of assigning a printer directly to a server, assign it to a group.
- Create a group record in the NDS. Assign a primary and a secondary server to each group.
- Create a print server record in the NDS. On it have a status of “Up” or “Down”.

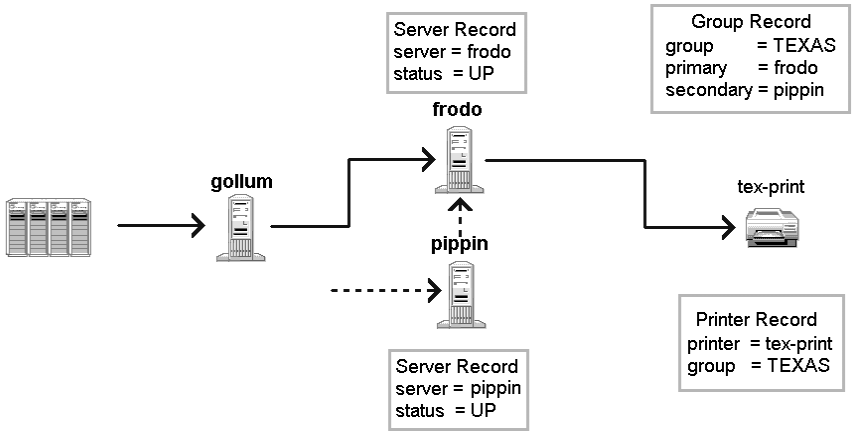
Print server decisions

- On receipt of print job server must:-
  - Lookup printer - get group code
  - Lookup group code - get primary & secondary print server
  - Lookup print servers - If primary server up, then use it, otherwise use secondary.
- Is "active" print server me? No, send job to active server.

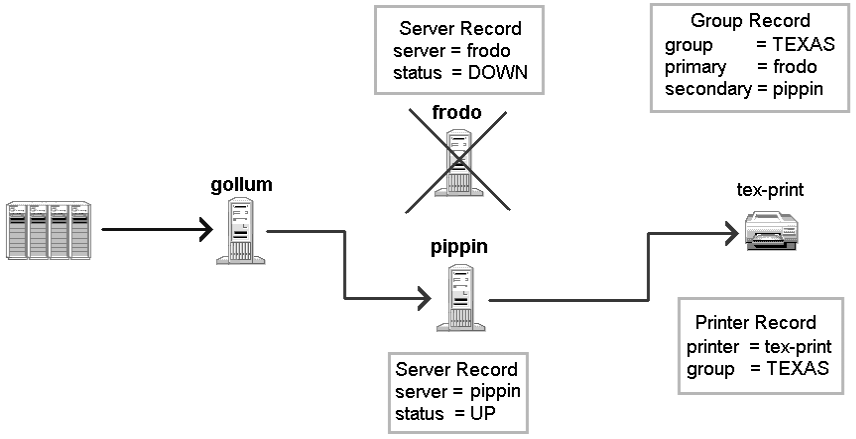
Fault tolerance in action

- If primary print server fails - mark it "down". Possibly automatically.
- All jobs going through primary print server, now go through secondary.

Normal Operation



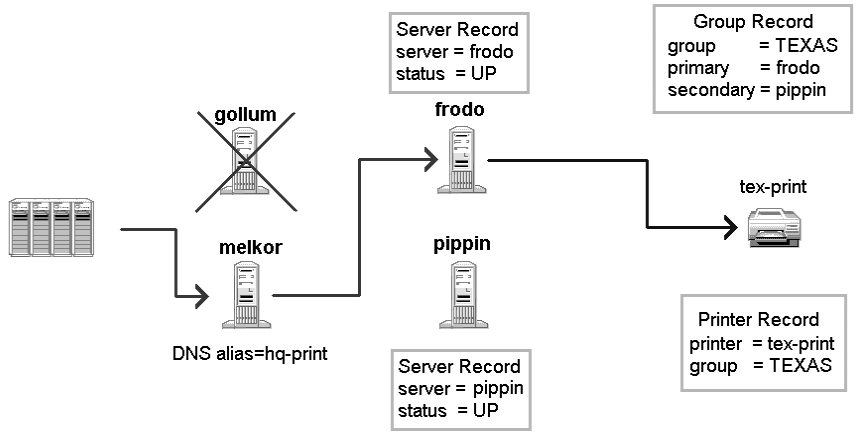
Failover In Operation



DNS is your friend

- What about incoming clients?
- Use DNS aliasing.
- Real name belongs to the machine.
- Alias belongs to the function.
- Good practice while you grow.
- Use NDS as dynamic DNS

Incoming server fails



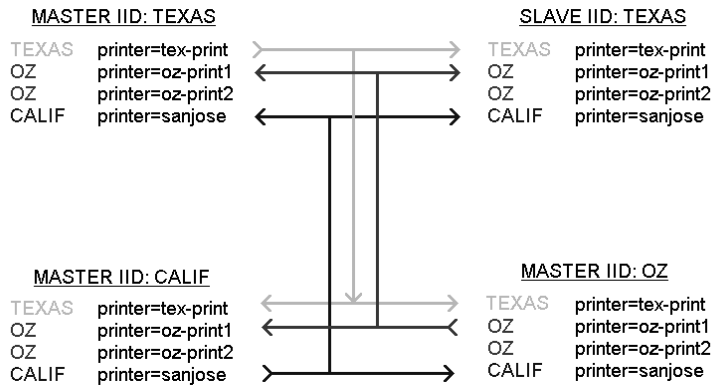
Distributed NDS

- NDS now critical to operation.
- Cannot allows NDS to be single point of failure
- NDS itself must be distributed.

SDDB

- Split records within each table according to region of "responsibility" - *iid*.
- Each record has an *iid* allocated.
- Each *iid* has a master server - responsible for all updates and propagation.
- Allow records to merge on all servers to form complete table again.

SDDB IID's



SDDB propagation

- All transactions done via fast (low overhead) UDP protocol.
- Only changed records propagated.
- Propagation quick - inside 20 seconds.

Summary

- Distributed systems are naturally fault tolerant.
- Distributed systems need a fast, distributed NDS to create "global machine".
- Distributed systems should be cheap.
- Linux fits this bill very well.

Future

- Many programs & protocols will need to be re-thought to work in a distributed machine.
- Linux is well placed to gain a significant foothold here - no-one else has come close.
- SDDB or another NDS will have to be become the core of all configuration files.